

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: Elektronické informační a řídicí systémy

**Návrh a realizace softwaru pro analýzu a
prezentaci log souborů**

**The Design and Impementation of software for
analysis and presentation of log files**

Bakalářská práce

Autor:	Adam Truhlář
Vedoucí práce:	Ing. Miroslav Holada, Ph.D.
Konzultant:	Ing. Josef Chaloupka, Ph.D.

V Liberci 21. 5. 2010

PROHLÁŠENÍ

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že s o u h l a s í m s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

PODĚKOVÁNÍ

Především bych rád poděkoval vedoucímu práce Ing. Miroslavu Holadovi, Ph.D. za poskytnuté informace a poskytnutý čas při vedení bakalářské práce. Dále bych chtěl poděkoval TRW Lucas Varity s.r.o. a Jaroslavu Mandíkovi za poskytnuté zdroje a možnost pracovat s webovými technologiemi. A nakonec bych chtěl poděkovat své rodině za podporu při studiu.

ABSTRAKT

Tato bakalářská práce je zaměřena na seznámení se s hlasovými technologiemi na pracovišti školitele a hlasovými aplikacemi, jejichž provoz je monitorován v „log” souborech. Bylo zapotřebí zmapování současných technologií, které se používají ke zpracování textových „log” souborů a jejich vyhodnocení např. pomocí grafického zobrazení ve formě grafů, či tabulek.

Cílem bylo navrhnout systém pro statistickou analýzu a online prezentaci „log” souborů s důrazem na informace o četnosti přístupu během dne, týdne, měsíce a roku. Další cíl spočíval ve zpracování zdroje dat, nebo-li vytvoření nadstavby hlasových aplikací, která zpracuje konkrétní „log” soubor pro následnou analýzu dat.

Klíčová slova: log soubor, aplikace, grafické zobrazení, prezentace, analýza.

ABSTRACT

This bachelor work is focused on identification of advisor voice technologies and voice applications on his working compartment. The application service is monitored via „log” files. It was necessary to become familiar with common technologies used for processing „log” files and it's graphical evaluation.

The task was to design statistical system for analysis and online presentation of „log” files with emphasis on information about daily, weekly, monthly, year access frequency statistics. Next goal was to process source of data or to create system upgrade of voice application to provide important data for statistic analyse.

Keywords: log file, application, graphical evaluation, presentation, analyse.

OBSAH

<u>PROHLÁŠENÍ</u>	<u>3</u>
<u>PODĚKOVÁNÍ</u>	<u>4</u>
<u>ABSTRAKT</u>	<u>5</u>
<u>ABSTRACT</u>	<u>6</u>
<u>SEZNAM POUŽITÝCH ZKRATEK</u>	<u>9</u>
<u>ÚVOD</u>	<u>11</u>
<u>1. HLASOVÉ TECHONOLOGIE</u>	<u>12</u>
<u>2. HLASOVÉ APLIKACE</u>	<u>14</u>
<u>3. ANALÝZA POŽADAVKŮ A NÁVRH SYSTÉMU</u>	<u>15</u>
3.1 ANALÝZA GENEROVANÝCH LOG SOUBORŮ	15
3.2 SOUČASNÉ TECHNOLOGIE PRO ANALÝZU	19
3.2.1 TABULKOVÉ PROCESORY	19
3.2.2 LOG ANALYZÁTORY	21
3.3 VOLBA VÝVOJOVÉHO PROSTŘEDÍ	23
<u>4. PREZENTAČNÍ TECHNOLOGIE</u>	<u>26</u>
4.1 HARDWAROVÉ VYBAVENÍ SERVERU	26
4.2 ZAPOJENÍ SERVERU DO POČÍTAČOVÉ SÍTĚ	27
4.3 OPERAČNÍ SYSTÉM	28
4.4 WEBOVÝ SERVER	29
4.5 TVORBA STRÁNEK PRO WEB SERVER	30
4.6 DATABÁZOVÝ SYSTÉM	39
<u>5 TECHNOLOGIE ZDROJE DAT</u>	<u>42</u>

<u>6. ZÁVĚR</u>	<u>46</u>
<u>SEZNAM POUŽITÉ LITERATURY</u>	<u>47</u>
<u>SEZNAM OBRÁZKŮ</u>	<u>48</u>
<u>PŘÍLOHY</u>	<u>49</u>
OBSAH DVD	49
SOFTWARE POTŘEBNÝ PRO SPUŠTENÍ APLIKACE	49
PODPOROVANÉ VERZE PROHLÍŽEČŮ	49

SEZNAM POUŽITÝCH ZKRATEK

PC	PERSONAL COMPUTER (OSOBNÍ POČÍTAČ)
IP ADRESA	INTERNET PROTOCOL ADDRESS (NUMERICKÝ POPIS ZAŘÍZENÍ, KTERÉ JE SOUČÁSTÍ POČÍTAČOVÉ SÍTĚ)
TCP	TRANSMISSION CONTROL PROTOCOL (PROTOKOL PRO PŘENOS TOKU BAJTŮ)
UDP	USER DATAGRAM PROTOCOL (PROTOKOL ZALOŽENÝ NA ODESÍLÁNÍ NEZÁVISLÝCH ZPRÁV)
IEFT	INTERNET ENGINEERING TASK FORCE (KOMISE TECHNIKY INTERNETU)
RFC	REQUEST FOR COMMENTS (ŽÁDOST O KOMENTÁŘE)
VoIP	VOICE OVER INTERNET PROTOCOL (DIGITÁLNÍ TELEFONIE PO INTERNETU)
OS	OPERATIONAL SYSTEM (OPERAČNÍ SYSTÉM)
GNU	PROJEKT SVOBODNÉHO SOFTWARE
LGPL	GNU LESSER GENERAL PUBLIC LICENSE (LICENCE SVOBODNÉHO SOFTWARE)
MB	MEGABYTE
GB	GIGABYTE
GHz	GIGAHERTZ
IIS	INTERNET INFORMATION SERVICES
BSD	BERKELEY SOFTWARE DISTRIBUTION (RODINA OS UNIX)
SQL	STRUCTURED QUERY LANGUAGE
GUI	GRAPHIC USER INTERFACE (GRAFICKÉ ROZHRANÍ)
API	APPLICATION PROGRAMMING INTERFACE
.NET	VÝVOJOVÁ PLATFORMA
ASP.NET	ACTIVE SERVER PAGES (SOUČÁSTÍ .NET)
MSSQL	MICROSOFT SQL SERVER

RAD	RAPID APPLICATION DEVELOPMENT
API	APPLICATION PROGRAMMING INTERFACE (ROZHRANÍ PRO PROGRAMOVÁNÍ APLIKACÍ)
C#	C SHARP (OBJEKTOVĚ ORIENTOVANÝ PROGRAMOVACÍ JAZYK)
CIL	COMMON INTERMEDIATE LANGUAGE (MEZI JAZYK PLATFORMY .NET)
CLI	COMMON LANGUAGE INFRASTRUCTURE (OTEVŘENÁ SPECIFIKACE POPISUJÍCÍ PROSŘEDÍ, KTERÉ POVOLUJE POUŽITÍ RŮZNÝCH PROGRAMOVACÍCH JAZYKŮ K POUŽITÍ NA ODLIŠNÝCH POČÍTAČOVÝCH PLATFORMÁCH)
C++	OBJEKTOVĚ ORIENTOVANÝ PROGRAMOVACÍ JAZYK
WWW	WORLD WIDE WEB („CELOSVĚTOVÁ PAVUČINA”)
HTTP	HYPertext TRAnsfer PROTOCOL
HDD	HARD DISK DRIVE (PAMĚŤOVÉ ÚLOŽIŠTĚ)
RAM	RANDOM-ACCESS MEMORY (PAMĚŤ S LIBOVOLNÝM PŘÍSTUPEM)
SCSI	SMALL COMPUTER SYSTEM INTERFACE (SADA PŘÍKAZŮ NA VÝMĚNU DAT MEZI ZAŘÍZENÍMI)
RDC	REMOTE DEKTOP CONNECTION (VZDÁLENÁ PLOCHA)
MSDN	MICROSOFT DEVELOPER PROGRAM NETWORK
CSV	COMMA-SEPARATED VALUES

ÚVOD

Přístupné hlasové aplikace na pracovišti vedoucího bakalářské práce již pracují delší dobu. Za tento časový úsek již vygenerovaly obrovské množství statistických dat ve formě tzv. „log” souborů. Pozn.: „Log” z angličtiny znamená poznámka, nebo záznam. Ve světě výpočetní technologie se jedná o často využívaný způsob jak uchovávat záznam akcí, které jednotlivé aplikace provedly. „Log” soubory jako takové jsou neocenitelným pomocníkem při řešení vyskytujících se problémů a to především z důvodu uchování své informace i po restartu PC nebo Serveru. Tato bakalářská práce se soustředí na informace o přístupu uživatelů k jednotlivým hlasovým aplikacím, které se v „log” souborech vyskytují. Protože každý uživatel je identifikován svoji IP adresou, lze z „log” souboru vyčíst informace o tom kolikrát se daný uživatel připojil ve zvoleném časovém intervalu, jak dlouho byl připojen, kolik různých uživatelů se celkem připojilo, nebo v kterých hodinách je aplikace nejvytíženější.

Bohužel záznamy v „log” souborech jsou jen těžko čitelné a s rostoucím obsahem informací to není jednodušší, proto je třeba „log” soubor podrobit analýze, která již poskytne přehlednější data.

Dalším problémem „log”souborů různých aplikací je jejich odlišné formátování, což znesnadňuje vývoj aplikací pro jejich analýzu. V konečném důsledku je často taková analýza na míru šitá dané aplikaci. Naštěstí hlasové aplikace na pracovišti školitele jsou napsané v programovacím jazyce C++ a k dispozici jsou i zdrojové kódy. Takže zde existuje prostor pro úpravu výstupu těchto aplikací, které nemusí jen zapisovat log soubor, ale mohou přímo zapisovat do databáze pro snadnější analýzu a prezentaci dat.

K zobrazení a prezentaci takto naměřených dat se nejčastěji používají tabulky, nebo především grafy, které jsou snáze zpracovatelné pro lidský mozek, než hromada řádků v textovém souboru.

Z výše uvedených poznatků lze rozdělit bakalářskou práci na dvě části. První částí je analytická a prezentační složka. Druhou část potom obsadí technologie zdroje dat.

1. HLASOVÉ TECHNOLOGIE

Hlasovými technologiemi rozumíme automatické zpracování řeči počítačem. Jedná se především o rozeznávání lidského hlasu a jeho převodu do digitalizované textové podoby a naopak převod textu na mluvené slovo.

Ústav Informačních technologií na Fakultě mechatroniky, informatiky a mezioborových studií zastřešuje hlasové technologie na Technické univerzitě v Liberci. Hlavní vědeckou aktivitou v tomto směru je automatické rozpoznávání řeči, které se těší stále větší oblibě mezi uživateli.

SpeechLab [7] je Laboratoř počítačového zpracování řeči na Technické univerzitě v Liberci. Byla založena v roce 1994. Počátkem 90. let byla pozornost laboratoře zaměřena na řešení problematiky rozpoznávání izolovaně pronášených slov a frází s postupným přechodem ke složitější úloze, kterou je rozpoznávání plynulé řeči. V současné době laboratoří vyvinuté systémy dokáží rozpoznat jednotlivá slova z rozsáhlých slovníků čítajících řádově stovky tisíc slov, a to od libovolného mluvčího, v reálném čase (do 1s) s úspěšností nad 95 %. Nejedná se jenom o aplikace pro PC ale i aplikace pro mobilní přístroje, s kterými se nejčastěji setkáváme. Např.: hlasové vytáčení je dnes již standardním vybavením většiny mobilních přístrojů. V dnešní době se jen těžko najde mobilní přístroj, který by hlasové rozpoznávání neumožňoval.

Při přepisování televizních pořadů se úspěšnost rozpoznávaných slov pohybuje okolo 90 % a u spojitého diktování libovolnou osobou (se slovníkem 350 000 slov). Rozpoznávání češtiny je velice náročná úloha, protože její slovník obsahuje na milióny různých slovních tvarů na rozdíl od často používané angličtiny, která je v tomto ohledu mnohem jednodušší.

Vývoj hlasových technologií, stejně jako vývoj všech ostatních technologií jde nezdělitelně kupředu. Setkáváme se s nimi stále častěji. Obklopují nás, stávají se součástí našeho života v nejrůznějších podobách. Ve výtahu nám oznámí v jakém patře se nacházíme, v tramvaji se dozvíme jaká je následující zastávka, popovídáme si s hlasovým automatem mobilního operátora, nebo handicapovaným usnadní každodenní úkony, které jsou pro zdravého člověka naprosto přirozené. Často si pokládám otázku kam až tyto technologie mohou dosáhnout, co mohou způsobit? V současnosti je systém rozpoznávání řeči testován jako náhrada soudního zapisovatele, což na jednu stranu ulehčí zapisovací proces, ale na druhou stranu, až technologie nebude vyžadovat

kontrolu lidského mozku, tak spousta lidí přijde o zaměstnání a nejedná se pouze o hlasové technologie. Mám z toho rozporuplný pocit. Jsem rád, že mohu být součástí technologického rozvoje, ale zároveň mám obavy z jeho využití, či zneužití.

2. HLASOVÉ APLIKACE

Na pracovišti školitele existuje několik veřejně přístupných hlasových aplikací, jejichž provoz je monitorován v „log“ souborech a z kterých lze zjistit informace o četnosti přístupu během určeného časového intervalu (den, týden, měsíc, rok, vlastní interval). Jedná se o následující aplikace:

- InfoCity - [8] Telefonní, informační a komunikační služba. Její výhoda spočívá v automatizaci (bez nutnosti lidské obsluhy) a v nenáročnosti na speciální telefonní přístroje. Umožňuje člověku, aby prostřednictvím telefonu získal informace, které mohou zajímat obyvatele či návštěvníky města Liberce. Pomocí hlasové komunikace s počítačem na druhém konci telefonního spojení tento systém poskytne informace např. o tom co se v daném týdnu hraje v libereckých divadlech.
- Dundis - Hlasový rozpoznávací systém, který rozpoznává mluvená izolovaná slova, slovní spojení a částečně i souvislou řeč. Právě na tomto systému je založen projekt InfoCity.
- Distribuované rozpoznávání řeči - Od klasického rozpoznávacího systému se liší v rozdělení aplikace od „rozpoznávače“, které spolu komunikují po síti. Výhodou takového řešení je nenáročnost aplikace na rozpoznávání a právě proto se systém distribuovaného rozpoznávání řeči využívá v zařízení jakými jsou např. PDA.

Výše uvedené aplikace jsou zájmem této bakalářské práce. Analýza jejich „log“ souborů poskytne informace o funkčnosti, či nefunkčnosti, využitelnosti a používání aplikací hlasových technologií.

3. ANALÝZA POŽADAVKŮ A NÁVRH SYSTÉMU

3.1 ANALÝZA GENEROVANÝCH LOG SOUBORŮ

```
(2009/18/05·08:53:16)·Client·(77.48.235.215:60997)·disconnected.¶
(2009/18/05·08:53:16)·New·client·accepted·from·77.48.235.215:65295.¶
(2009/18/05·08:54:29)·Client·(77.48.235.215:65295)·disconnected.¶
(2009/18/05·13:57:07)·-----¶
(2009/18/05·13:57:07)·Starting·up·the·DSR·server·thread...¶
(2009/18/05·13:57:07)·DSR·server·thread·running.¶
(2009/18/05·13:57:07)·Setting·up·the·DSR·server...¶
(2009/18/05·13:57:07)·Client·server·side·on-line,·using·address·and·port.¶
(2009/18/05·13:57:13)·Starting·up·the·RFM·thread...¶
(2009/18/05·13:57:13)·RF·Master·thread·running.¶
(2009/18/05·13:57:13)·Setting·up·the·RFM·service...¶
(2009/18/05·13:57:13)·Master·on-line,·using·address·and·port·147.230.81.1¶
(2009/18/05·13:57:13)·RF·Master·on-line.¶
(2009/18/05·13:57:13)·Server·on-line.¶
(2009/18/05·13:57:15)·New·slave·accepted·from·147.230.81.176:1034.¶
(2009/18/05·13:57:15)·New·slave·accepted·from·147.230.81.176:1035.¶
(2009/18/05·14:51:44)·New·client·accepted·from·147.230.81.240:53592.¶
(2009/18/05·15:49:05)·Client·(147.230.81.240:53592)·disconnected.¶
```

Obrázek 1 Ukázka log souboru ze serveru Dundis

Výše ukázaný log soubor má přes 3 milióny řádek záznamů. Vyčíst důležité statistické informace o denním, měsíčním, nebo ročním přístupu je na první pohled pro člověka nemožné. Na řadu tedy přichází analýza. Drtivá většina log souborů různých aplikací má však podobnou strukturu. Jeden záznam totiž obsahuje čas události a název události samotné. Např.: *(2009/18/05 13:57:07) DSR server thread running*. Informuje administrátora o tom, že 18.5.2009 v 13:57:07 běželo vlákno DSR serveru. V takto jednoduchém záznamu je hned několik úskalí:

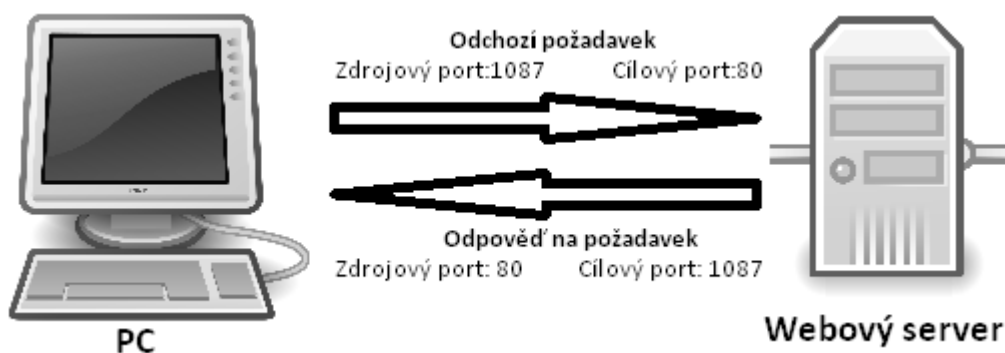
- Formátování času - Po celém světě se používá několik různých formátů času. Při špatné volbě formátu může často dojít k výjimkám, či chybám v programu.
- Text události - Jedinečný téměř pro každou aplikaci.
- Oddělení jednotlivých parametrů v jednom záznamu.

Pro potřeby jednotného systému na analýzu a prezentaci dat je zapotřebí veškeré parametry sjednotit. Formátování času musí být ve všech záznamech shodné, stejně jako texty událostí generované danou aplikací. Protože se tato práce soustředí na informace o četnosti přístupu uživatelů, lze vynechat všechny záznamy, které neobsahují žádné informace o přístupu uživatelů k aplikaci. V případě log souboru z aplikace Dundis server se jedná o záznamy obsahující text „*New client accepted from xxx.xxx.xxx.xxx:yyyy*” a „*Client (xxx.xxx.xxx.xxx:yyyy) disconnected*”. První text obsahuje informaci o připojení klienta k aplikaci a druhý informuje o jeho odpojení. Identifikace klienta lze odhalit z kombinace IP adresy (xxx.xxx.xxx.xxx v textu) a Portu.(yyyy v textu).

Vysvětlení IP:

- [9] IP (z angličtiny Internet Protocol) Datový protokol pro přenos přes paketové síť. Jedná se o základní protokol dnešního internetu. Data se v IP síti posílají po blocích nazývaných datagramy. Jednotlivé datagramy putují sítí zcela nezávisle. IP v doručování datagramů poskytuje nespolehlivou službu, tj. všechny zařízení na trase se datagram snaží podle svých možností poslat blíže k cíli, ale nezaručují jeho doručení. K adresování síťového zařízení se využívá IP adresy.
- [10] IP adresa je v informatice číslo, které jednoznačně identifikuje síťové zařízení v počítačové síti, které používá IP (internetový protokol). V současné době je nejrozšířenější verze IPv4, která používá 32bitové adresy zapsané dekadicky po jednotlivých oktetech (osmicích bitů), například 192.168.0.1. Slouží k rozlišení zařízení (PC uživatele) připojených k počítačové síti. Pomocí IP (Internet Protocol) spolu komunikují všechna zařízení v internetu. IP adresa v log záznamu bývá zpravidla doprovázena portem.

- [11] Síťový port je speciální číslo (0 až 65535), které slouží v počítačových sítích při komunikaci zařízení pomocí protokolů TCP a UDP k rozlišení aplikace v rámci počítače.



Obrázek 2 Komunikace mezi PC a Serverem

- TCP [12] (Transmission Control Protocol) Jeden ze základních protokolů sady IP. Konkrétně představuje transportní vrstvu. Použitím TCP mohou aplikace na počítačích propojených do sítě vytvořit mezi sebou spojení, přes které se přenáší data. Protokol garantuje spolehlivé doručování a doručování ve správném pořadí. TCP také rozlišuje data pro vícenásobné, současně běžící aplikace. V této bakalářské práci využiji protokol TCP pro webovou aplikaci. Dokumentace k protokolu TCP lze nalézt v IEFT RFC 768 na adrese <http://www.ietf.org/rfc/rfc768.txt>.
- [13] UDP (User Datagram Protocol) Protokol z transportní vrstvy IP. Na rozdíl od TCP protokolu nedává záruky na datagramy, které přenáší mezi zařízeními v počítačové síti. Protokol UDP nezaručuje, zda se přenášený datagram neztratí, zda se nezmění pořadí doručených datagramů nebo zda se některý datagram nedoručí vícekrát. Bezstavovost UDP protokolu je užitečná pro servery, které obsahují mnoho klientů nebo pro nasazení kde se počítá se ztrátami datagramů a není vhodné, aby se ztrácel čas novým odesíláním nedoručených zpráv. Využití UDP protokol nalézá především v VoIP (telefonování prostřednictvím internetu), online hrách, nebo při komunikaci s databázovým systémem, který využiji pro analýzu a prezentaci log záznamů. Zdokumentovaný UDP protokol, lze nalézt pod označením IETF RFC 793 na adrese www.ietf.org/rfc/rfc793.

K vyčtení statistických dat z log záznamu je zapotřebí volba vhodné struktury, která výrazně zjednoduší další analýzu dat. Pro systém analýzy četnosti přístupů jsem si zvolil tabulku.

ČAS	ÚDÁLOST	IP	PORT
2008/16/09 07:49:07	PŘIPOJENÍ KLIENTA	147.230.81.178	1041
2008/16/09 13:33:07	ODPOJENÍ KLIENTA	147.230.81.178	1041
2008/23/09 08:35:19	PŘIPOJENÍ KLIENTA	147.230.81.178	1047
2008/23/09 13:09:56	PŘIPOJENÍ KLIENTA	82.128.191.199	2159
2008/23/09 13:11:00	ODPOJENÍ KLIENTA	147.230.81.178	1047

Tabulka 1 Ukázka log záznamu ve formě tabulky

Takto může vypadat jednoduchá tabulka, z které se dají zjistit informace o četnosti přístupů klientů v časovém horizontu k aplikaci. Připojení klientů se neuskutečňuje na stejném portu. Každý současně připojený uživatel musí mít svůj vlastní port. O dynamické přiřazování portů se stará server, který na žádost připojení klienta přiřadí port, pomocí kterého klient se serverem komunikuje. Výběr portů probíhá z nastavené skupiny volných portů. Dále lze z Tabulky 1 zjistit, že mezi připojením a odpojením jednoho klienta, může být i několik dalších záznamů jiných klientů o připojení a odpojení. Tento fakt komplikuje zjištění délky připojení jednotlivých klientů. Modifikací Tabulky 1 lze docílit zjištění délky připojení.

ČAS UDÁLOSTI	ČAS KONCE UDÁLOSTI	UDÁLOST	IP	PORT
2008/16/09 07:49:07	2008/16/09 13:33:07	PŘIPOJENÍ/ODPOJENÍ KLIENTA	147.230.81.178	1041
2008/23/09 08:35:19	2008/23/09 13:11:00	PŘIPOJENÍ/ODPOJENÍ KLIENTA	147.230.81.178	1047
2008/23/09 13:09:56		PŘIPOJENÍ/ODPOJENÍ KLIENTA	82.128.191.199	2159

Tabulka 2 Upravená tabulka pro zjištění délky připojení klienta

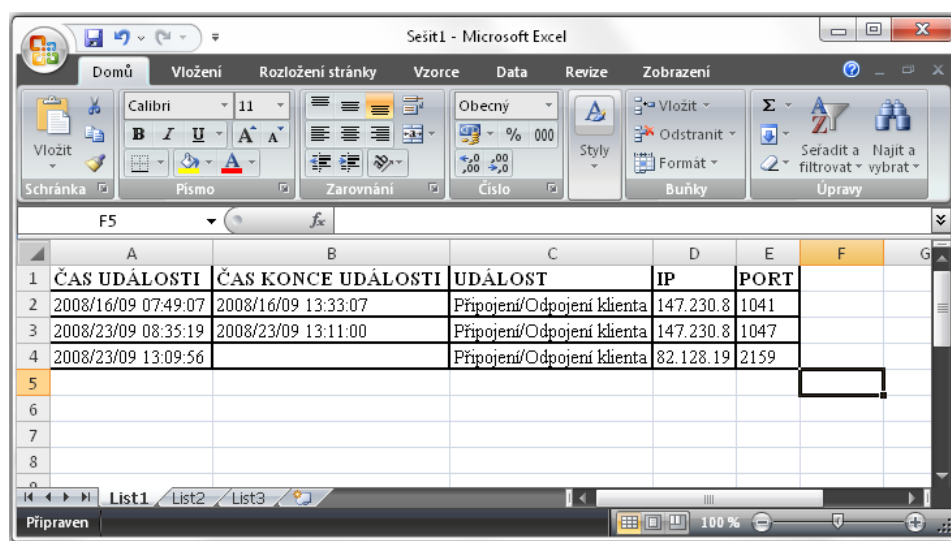
Upravení Tabulky 1 může vypadat např. jako Tabulka 2. Délka připojení se potom spočítá takto: ČAS KONCE UDÁLOSTI - ČAS UDÁLOSTI = DÉLKA UDÁLOSTI. Jediný problém k vytvoření Tabulky 2 spočívá ve spárování připojení a odpojení klienta. Záznamy v log souborech nemusí být 100 % spolehlivé. Stává se tak v případě výpadku serveru, kdy se ukončí veškerá připojení a tato připojení potom nejsou zaznamenána do log souboru. Při párování takových záznamů potom dojde k situaci, kdy nelze nalézt odpovídající odpojení klienta. Záznam v tabulce je poté nekompletní, jako je zobrazeno v Tabulce 2 na 3. řádku. Otázkou zůstává jak s takovýmto záznamem naložit při vyhodnocení? Nakonec byla zvolena varianta, která s neúplným záznamem nepočítá při zjišťování statistik souvisejících s délkou připojení.

3.2 SOUČASNÉ TECHNOLOGIE PRO ANALÝZU

3.2.1 TABULKOVÉ PROCESORY

První věc, která přijde na mysl ve spojitosti s tabulkami, jsou tabulkové procesory. [3] Tabulkový procesor (anglicky spreadsheet) je program zpracovávající tabulku informací (matice informací). V jednotlivých buňkách tabulkového procesoru mohou být uložena data či vzorce počítající s těmito daty. V tom případě se v tabulce zobrazují data vypočtená ze vzorců. Tabulkový procesor prvotní uplatnění nacházel zejména ve finančnictví a proto obsahoval zejména funkce vhodné k finančním výsledkům. Dnešní tabulkové procesory mají mnohem širší záběr funkcionality. Využití v souvislosti s tabulkou log záznamů se tu přímo nabízí. V současnosti existuje několik produktů tabulkových procesorů:

- Microsoft Excel - Součástí placeného balíčku Microsoft Office, určeného výhradně pro OS (Operační Systém) Microsoft Windows. Nejnovější verze 2010 se soustředí na sdílení dokumentů online. Po vzoru Google Apps. bude existovat i webová podoba Office. Přichází s 64bitovou podporou pro práci s většími soubory a pro přesnější výpočty.



Obrázek 3 Aplikace Microsoft Office 2007

- OpenOffice.org Calc - Z kancelářského balíčku OpenOffice společnosti Sun Microsystems. Jedná se o bezplatnou alternativu k aplikaci Microsoft Excel šířenou pod licencí LGPL. Podporován většinou současných OS.

Analýza log záznamů by v případě tabulkových procesorů probíhala ve třech krocích:

1. Načtení dat do tabulkového procesoru
2. Vytvoření funkcí pro analýzu dat
3. Naformátování výstupních dat např. do grafu

Už v prvním kroku nastává největší problém, spojený s omezeními tabulkových procesorů jak ilustruje následující tabulka.

VLASTNOST	OMEZENÍ		
	EXCEL 2003	EXCEL 2007	OPENOFFICE 3.0 CALC
POČET ŘÁDKŮ NA LISTU	65536	1048576	65536
POČET SLOUPCŮ NA LISTU	255	16384	1024
POČET LISTŮ	OMEZENO PAMĚTÍ	OMEZENO PAMĚTÍ	256

Tabulka 3 Omezení tabulkových procesorů

Log soubor z aplikace Dundis Server má více než 270 MB a téměř na 4 milióny řádků. Jeden záznam logu odpovídá právě jednomu řádku v log souboru. I kdyby byl použit Excel 2007 pro analýzu dat, tak bude maximálně obsahovat 1/4 všech informací oproti log souboru. I když velkou část dat log souboru tvoří chybové hlášení a zkrácený soubor by se mohl celý načíst do tabulkového procesoru, práce s takovým množstvím dat je náročná na výkon a tedy neuvěřitelně pomalá. Jenom otevření souboru tabulkového procesoru, který obsahuje na 65000 řádek informací může trvat několik minut v závislosti na výkonu počítače. Čím větší množství dat tabulka obsahuje, tím jsou výpočty nad těmito daty náročnější. Z výše uvedených fakt logicky vyplývá nevhodnost použití tabulkového procesoru na analýzu log záznamů.

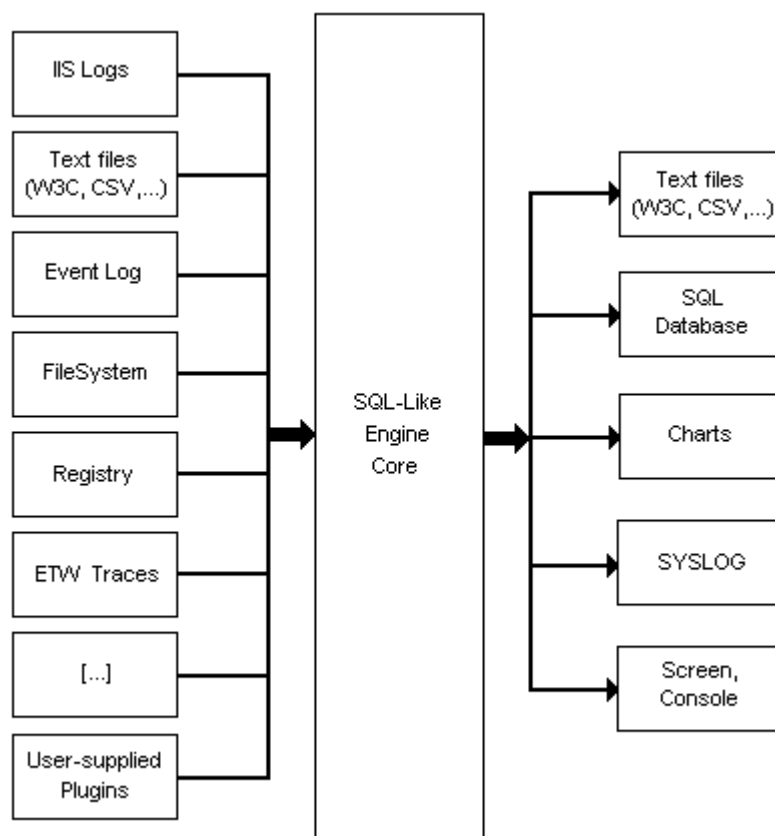
3.2.2 LOG ANALYZÁTORY

Software pro analýzu log souborů se nazývá log analyzátor. Vybírat lze z nepřeberného množství placených nástrojů, které vygenerují nádherné statistiky, ale pouze z předem nadefinovaných zdrojů. Tyto zdroje jsou z pravidla web servery:

- IIS - (Internet Information Service) Webový server pro platformu OS Windows.
- Apache - Webový server s otevřeným kódem dostupný pro většinu všech dostupných systémů: BSD, Linux, Solaris, Mac OS X a Windows. Nejpoužívanější v současné době.

Jedná se tedy o jednoúčelové nástroje pro analýzu web serverů zpravidla bez možnosti načtení log souborů s vlastním formátováním. Odstraněním množiny všech placených nástrojů se výběr značně zúží.

LogParser 2.2 je utilita z dílen firmy Microsoft. [14] Umí analyzovat rozdílné typy „log“ souborů a souborových formátů. LogParser je nástroj ovládaný pomocí příkazové řádky. Mimo jiné umožňuje analyzovat Protokol událostí ve Windows. Jak tedy LogParser funguje? Jak lze zjistit z následujícího obrázku jeho architektury, LogParser umožňuje analyzovat log soubory v mnoha formátech jako jsou například textové soubory, protokoly událostí nebo registry. K analýze využívá systém podobný SQL (dotazovací jazyk používaný v relačních databázích). Data se poté získávají pomocí dotazů podobných SQL.



Obrázek 4 Architektura LogParseru, převzato z
 <<http://www.codinghorror.com/blog/2005/08/microsoft-logparser.html>>

LogParser umožňuje několik možných výstupů:

- Textové soubory
- SQL databáze
- Grafy
- SYSLOG - standart pro logování zpráv programů
- Konzole

Pro lidi kteří preferují jiný způsob ovládání, než z okna konzole je k dispozici několik GUI (grafické rozhraní), které obsahují základní ovládací prvky. Bohužel i přes robustnost toho nástroje se nepovedlo správně načíst soubor pro potřebný výstup.

Najít další volně dostupné log analyzátoři bylo velice složité. Poslední nalezený kandidát je Analog. Jenomže Analog se orientuje na web servery a jeho vývoj už byl dávno ukončen. Na internetu existuje nepřeberné množství aplikací, nástrojů. Prohledat všechny dostupné aplikace na analýzu log souborů a vyzkoušet jestli vyhovují všem

požadavkům práce není v mých silách. Je vcelku pravděpodobné, že ze všech možných programů jsem nějaký přehlédl, takže budu rád za případné doplnění informací. Nezbyvá tedy nic jiného, než si systém analýzy a prezentace naprogramovat sám.

3.3 VOLBA VÝVOJOVÉHO PROSTŘEDÍ

V současné době existuje nepřehledné množství programovacích jazyků, prostředí, které soupeří o místo nejrozšířenější platformy pro psaní programů. Mezi nejdůležitější vlastnosti programovacího jazyka patří přenositelnost, robustnost, podpora ze strany vývojářů a v neposlední řadě zkušenosti programátorů s daným jazykem. V závislosti na robustnosti dané platformy lze log soubor nejen zpracovat, ale i výsledek uložit např. do XML souboru nebo databáze a zároveň vytvořit webovou aplikaci, která se postará o zpřístupnění statistik z log souboru online.

K dispozici je hned několik vývojových prostředí. Jedním z nejrozšířenějších představitelů je Microsoft .NET. Jedná se o soubor technologií v softwarových produktech, které tvoří celou platformu dostupnou nejen pro Web, Windows i Pocket PC. Základní komponentou je .NET Framework, prostředí potřebné pro běh aplikací a nabízející jak spouštěcí rozhraní, tak potřebné knihovny. Pro vývoj .NET aplikací vydal Microsoft Visual Studio .NET. V současné verzi 2008 umožňuje vytvářet aplikace všeho druhu. Od webových aplikací až po grafické aplikace využívající DirectX. Existuje GNU obdoba .NET, která se nazývá DotGNU (GNU je projekt zaměřený na svobodný software). DotGNU se stará o přenositelnost celé platformy, tedy umožňuje spouštět všechny .NET aplikace na unixových platformách (Linux, Mac OS X, Solaris). Programátor píšící .NET aplikace není omezen volbou jednoho programovacího jazyka. Bez ohledu na to v čem byla aplikace napsána, se vždy přeloží do mezijazyka Common Intermediate Language. Možností výběru programovacího jazyka je hned několik. Od jednoduchého VB.NET (následovník Visual Basicu), přes C# (podobný pro změnu Java) až po Managed C++. Pro vývoj webových aplikací Microsoft připravil technologii ASP.NET, kterou lze snadno použít na online grafické zobrazení výstupu zpracovaného log souboru.

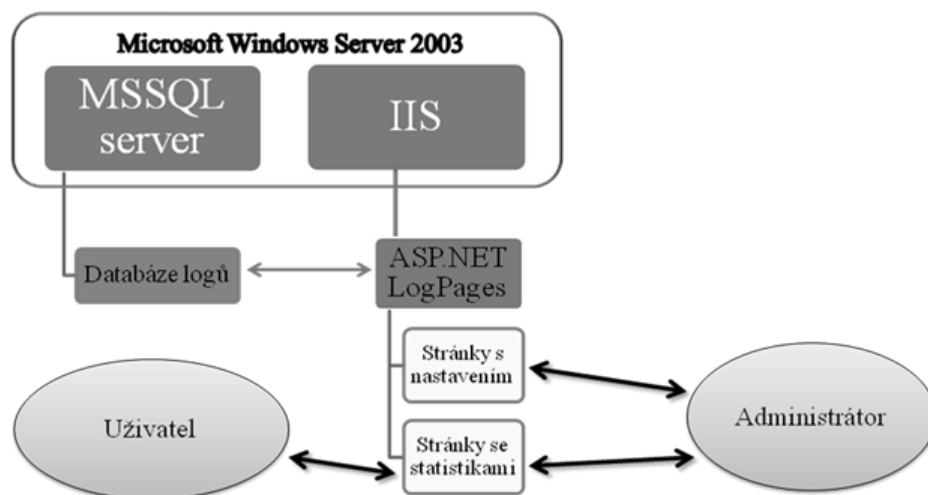
Do další rozšířené platformy lze zařadit platformu Java. K dnešnímu dni se Java rozděluje na další dílčí platformy:

- JavaCard – aplikace provozované v rámci „Smart Card“ (platební karty)
- Java ME – aplikace pro mobilní zařízení
- Java SE – aplikace pro stolní počítače
- Java EE – rozsáhlé aplikační systémy

Výše uvedené platformy sdílejí syntaxi jazyka Java, virtuální stroj Javy (stará se o přenositelnost na různé platformy), obdobné API standardních knihoven. Java je open source objektově orientovaný jazyk. Na platformě Java lze bez větších problémů zpracovat i tuto práci.

Jako poslední alternativu uvedu Delphi. Jedná se o integrované vývojové prostředí firmy Borland sloužící pro tvorbu aplikací výhradně na platformě Microsoft Windows. Znalost Object Pascalu je základním předpokladem pro tvorbu aplikací v prostředí Delphi. Obsahuje RAD (Rapid Application Development), který umožňuje grafický návrh uživatelského rozhraní, na jehož základě je vytvářena kostra zdrojového kódu. Podobně jako ve Visual Studiu je programování v Delphi založeno na použití komponent. Vývoj aplikací pomocí komponent výrazně usnadňuje jejich vytváření. Delphi je založen na programovacím jazyce Pascal. Umožňuje pracovat s databázemi. Bohužel vývoj v Delphi má několik nevýhod. Jednak tato platforma není přenosná, není zadarmo a v neposlední řadě kód není příliš optimalizovaný.

Volba prezentační technologie pro mě nebyla příliš složitá vzhledem k tomu, že již od malička chtěl nechtěl jsem obklopován technologiemi Microsoftu. MS-DOSem počínaje a Visual Studiem konče. Nejtěžším na celém projektu byla otázka kde a jak začít? S požadavkem na online zobrazení výsledků má volba padla na webovou aplikaci. Microsoft k tomuto účelu vytvořil technologii ASP.NET, která splňuje požadavek online přístupných výsledků a protože se jedná o technologii v .NET Frameworku, tak snadno pracuje s databázemi. Skutečnost že ASP.NET komunikuje s databázemi ulehčila spoustu práce při zpracování a ukládání log souborů. Pro vývoj webové aplikace jsem použil Microsoft Visual Studio 2008 (ASP.NET stránky za pomoci programovacího jazyka C#). Microsoft SQL server poskytl databázi a webový server pro hostování ASP.NET webové aplikace zastoupí IIS (Internetová informační služba, která je již integrována v instalaci Windows).



Obrázek 3 Schéma prezentační technologie

Jak to celé funguje? Na serveru s operačním systémem Windows Server 2003 je nainstalován nezbytný software:

- .NET Framework 1.0, 2.0, 3.5 SP1
- Microsoft SQL server 2005
- IIS (Internet information Services)

Vytvořenou webovou aplikaci jsem pojmenoval LogPages. LogPages jsou umístěny na lokálním disku serveru. V IIS je nastavena cesta, přístupová práva a další důležitá nastavení k těmto stránkám.

4. PREZentační technologie

Prezentační technologie bude předkládat, ukazovat výsledky analýzy log záznamů. Má-li být k dispozici online (k dispozici pro více uživatelů v počítačové síti), nezbyvá nic jiného, než využít technologii WWW (aplikace internetového protokolu HTTP) využívané po celém světě k prezentaci informací. Pro vytvoření webové aplikace a její nasazení do provozu je zapotřebí splnit několik základních bodů:

- Přístup k hardwarovému vybavení (počítač, server)
- Zapojení serveru do počítačové sítě
- Operační systém
- Webový server pro prezentaci informací
- Databázový systém pro úschovu a analýzu dat
- Nastavení počítače, které umožní komunikaci mezi klientem a serverem

4.1 HARDWAROVÉ VYBAVENÍ SERVERU

K dispozici jsem dostal přístup k serveru Dundis umístěném ve vedlejší místnosti pracoviště školitele. Webové aplikace nejsou náročné na výpočetní výkon, ale záleží zde na počtu klientů, které musí server obsloužit. Platí zde přímá úměrnost: Čím více připojených klientů a aplikací běžících na serveru, tím větší výpočetní výkon je potřeba. Výpočetní výkon serveru lze odvodit z jeho hardwarového vybavení. Jak je na tom Dundis server s hardwarovým vybavením a výpočetním výkonem?

POČET PROCESORŮ (CPU)	2
TYP CPU	INTEL XEON (PRESTONIA)
VÝROBNÍ TECHNOLOGIE CPU	130 NM
TAKT CPU	2,8 GHz
POČET SOUČASNĚ ZPRACOVÁVANÝCH VLÁKEN	2 VLÁKNA NA PROCESOR, 4 VLÁKNA CELKEM
VELIKOST OPERAČNÍ PAMĚTI RAM	2816 MB
VELIKOST ÚLOŽNÉHO PROSTORU	33,91 GB
ŘADIČ PŘIPOJENÍ DISKŮ	SCSI

Tabulka 4 Parametry Dundis serveru

Kódové označení procesoru „Prestonia” se poprvé objevuje v roce 2002. Dundis server tedy neobsahuje nejmodernější komponenty, ale jeho výpočetní výkon je více než dostatečný na provozování web serveru. S roustoucím stářím serveru roste i možnost selhání pevného disku.

4.2 ZAPOJENÍ SERVERU DO POČÍTAČOVÉ SÍTĚ

Server je zapojen do univerzitní sítě LIANE (počítačová síť technické univerzity v Liberci, která pokrývá veškeré hlavní budovy univerzity) a využíván několika uživateli zároveň. Připojit se k němu lze nejenom v rámci LIANE, ale i v rámci internetu, což činí server snadno dostupným. Pro práci na serveru není za potřeby používat periferie (klávesnice, myš) zapojené přímo do serveru. Správa a ovládání serveru probíhá pomocí vzdálené plochy (RDC).

Připojení ke vzdálené ploše vyžaduje vyplnění údajů:

- IP adresu počítače nebo serveru ke kterému se chceme připojit.
- Uživatelské jméno.
- Heslo pro vyplněné uživatelské jméno.

Předpoklady pro úspěšné připojení ke vzdálené ploše:

- Klient i server běží na platformě Microsoft Windows.
- Služba vzdálené plochy je kompatibilní s připojením ke vzdálené ploše
- Na serveru je spuštěna služba vzdálené plochy.
- Server a klient umožňuje vzájemné spojení.
- Klient, který se chce připojit zná přihlašovací údaje na server.
- Není překročen limit současně připojených klientů.

Práce se vzdálenou plochou se nijak neliší od práce v přítomnosti počítače. Pokud je okno aplikace vzdálené plochy aktivní, tak veškerý pohyb myši, jakékoliv stisknutí tlačítka se na serveru vyhodnocuje, jako v přítomnosti uživatele, který by

ovládal server pomocí klávesnice a myši zapojené přímo do serveru. V možnostech vzdálené plochy je i sdílení lokálních zdrojů (tiskárny, pevné disky a další), které ulehčují manipulaci se soubory při nahrávání i stahování.

4.3 OPERAČNÍ SYSTÉM

Jak již bylo uvedeno výše, operační systém je nezbytnou součástí každého počítače. Jedná se o základní programové vybavení počítače, prostřednictvím kterého uživatel komunikuje s hardwarem. Od uvedení prvních operačních systému uplynulo již mnoho let. Za tuto dobu prošly operační systémy mnoha změnami až do současnosti kdy se funkcionality nedá srovnávat s prvními předchůdci. Současná nabídka operačních systému lze rozdělit do 3 platforem:

- GNU / Linux
- Unix
- Windows

GNU (z angličtiny „pakůň“) je maskotem projektu GNU zaměřeného na svobodný software. Z toho vyplývá že GNU/Linux není placeným operačním systémem. Základy GNU/Linux sahají k jeho předchůdci Unix. Jedná se o moderní operační systém splňující náročné požadavky firemních serverů, stejně jako požadavky osobního počítače s vysokým zabezpečením.

Platformu Unix zastupují operační systémy pod označením Solaris vyvíjené společností Sun Microsystems. V poslední dostupné verzi Solaris 10 obsahuje podporu pro architekturu x86-x64. Solaris se vyznačuje stabilitou a robustností, kvůli které je nasazován do serverů s velkým množstvím procesorů. Stejně jako Linux lze Solaris používat bezplatně a to i v komerční sféře. Zpoplatněná je až technická podpora.

Skupinu současných operačních systému uzavírá placená platforma Microsoft Windows. Na poli osobních počítačů nejrozšířenější platforma, poskytuje celou škálu nejrůznějších programů. Vyhovuje potřebám jak domácnostem tak společnostem s mnoha pobočkami. Domácí využití a pracovní stanice pokrývá produktová řada Windows XP, Windows Vista a nejnovější Windows 7. Na servery se nasazuje Windows Server 2003, nebo Windows Server 2008, které oproti svým protějškům pro domácí využití nabízí další funkce, služby a podpory, ale hlavně upravují některá omezení domácích protějšků.

Volba operačního systému na serveru Dundis byla již vyřešená, protože server spoléhá na operační systém Microsoft Windows Server 2003 (32bit) ve verzi Standard, z kterého vyplívají následující omezení:

- Podpora maximálně 4 procesorových jader
- Podpora maximálně 4 GB operační paměti

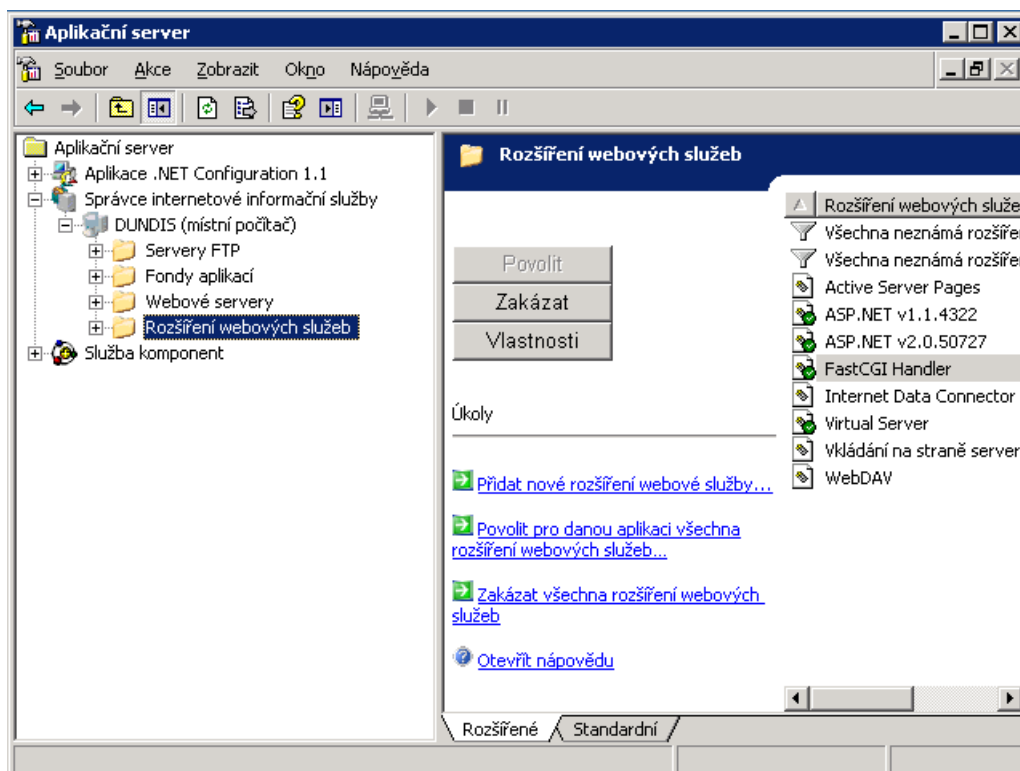
4.4 WEBOVÝ SERVER

Na platformě Microsoft Windows je webový server IIS (internetová informační služba) pevně spojena s verzí operačního systému. Podrobnější informace o jednotlivých verzích IIS lze nalézt v následující tabulce

VERZE OPERAČNÍ SYSTÉMU	VERZE IIS
WINDOWS 2000	IIS 5.0
WINDOWS XP	IIS 5.1
WINDOWS SERVER 2003	IIS 6.0
WINDOWS VISTA	IIS 7.0
WINDOWS 7	IIS 7.5
WINDOWS SERVER 2008	IIS 7.0
WINDOWS SERVER 2008 R2	IIS 7.5

Tabulka 5 Verze IIS a Windows

IIS 5.1 je naprosto nepoužitelná pro širší nasazení, kvůli omezení počtu současně připojených klientů, kterým je hodnota 10. IIS od verze 6.0 už tímto neduhem netrpí a přináší další vylepšení. Uživatel platformy Windows nemusí využívat služeb integrovaného web serveru, protože existuje alternativa v podobě open source webového serveru Apache HTTP Server, který je nejpopulárnějším na internetu.



Obrázek 4 Správa IIS 6.0

4.5 TVORBA STRÁNEK PRO WEB SERVER

Využito bylo technologie ASP.NET v kombinaci CLR podporujícím jazykem C#. Vývoj stránek Log pages probíhal ve vývojovém prostředí Microsoft Visual Studio 2008, které není zadarmo avšak na webu Microsoftu lze stáhnout Visual Web Developer 2008 Express, který je odlehčenou a bezplatnou variantou Visual Studia. Vývoj ASP.NET webových aplikací za pomoci Visual Studia je překvapivě jednoduchý. Vývojář má k dispozici oficiální stránky ASP.NET (www.asp.net), které obsahují nepřehledné množství návodů, stejně jako MSDN (msdn.microsoft.com) online knihovnu, v které najdeme základní zdroje informací pro programátory nejenom .NET aplikací. Nesmím také zapomenout na početnou komunitu lidí na nejrůznějších diskusních fórech, která ráda poradí, pomůže nebo vysvětlí programátorův vyskytnuvší problém.

.NET stránky (oficiálně známé jako web forms) jsou obsaženy v souborech s příponou „.aspx“, které obsahují statický XHTML značkovací jazyk společně s jazykem

obsahujícím server-side (na straně serveru) controls (ovládací prvky), kde vývojáři umísťují veškerý statický i dynamický obsah webové stránky. Popřípadě dynamický kód, který běží na serveru může být umístěn do bloku „<% -dynamický kód-%>”.

Pro větší přehlednost kódu slouží „Code-behind” model, kdy je stránka s dynamickým kódem umístěna do vlastního souboru s příponou „.vb nebo .cs” v závislosti použitého jazyka. Soubor s dynamickým kódem poté musí být přilinkován k „.aspx” souboru pomocí atributu „CodeFile” viz. Obrázek 5.

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="Settings.aspx.cs" Inherits="Settings" Title="LOG-Settings" %>
```

Obrázek 5 Přilinkování modelu Code-Behind v souboru Settings.aspx

```
using ...

public partial class Settings : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack == false)
        {
            //Info.Text = "";
        }
    }
}
```

Obrázek 6 Ukázka souboru Settings.aspx.cs s dynamickým kódem

Kromě souborů s příponou „.aspx” nebo „.vb nebo .cs” jsou s ASP.NET spojeny i další soubory s příponami, které názorně ukazuje následující Tabulka 6.

PŘÍPONA SOUBORU	POPIS
ASAX	(GLOBAL.ASAX) VOLITELNÝ SOUBOR UŽÍVANÝ PRO DEKLARACI A ŘÍZENÍ APLIKAČNÍCH UDÁLOSTÍ
ASCX	WEBOVÝ UŽIVATELSKÝ OVLÁDACÍ PRVEK (USER CONTROL)
ASHX	VLASTNÍ HTTP HANDLERY (PROCES KTERÝ JE

	SPUŠTEN JAKO ODPOVĚD NA POŽADAVEK)
ASMX	STRÁNKA S WEB SLUŽBOU
AXD	WEB HANDLER
BROWSER	SCHOPNOSTI PROHLÍŽEČŮ, SLOUŽÍ K OPTIMALIZACI KÓDU PRO RŮZNÉ PROHLÍŽEČE
CONFIG	(WEB.CONFIG) UCHOVÁVÁ NASTAVENÍ PRO DANOU WEBOVOU APLIKACI
DBML	LINQ TO SQL (OBJEKTOVĚ-RELAČNÍ MAPPER) ALTERNATIVA K POUŽÍVÁNÍ T-SQL DOTAZŮ
MASTER	ZNAČÍ MASTER PAGE (VYSVĚTLENO NÍŽE)
RESX	ZDROJOVÉ SOUBORY PRO VÍCE JAZYKOVÉ STRÁNKY
SITEMAP	MAPA STRÁNEK VYUŽÍVÁNA PŘI UMÍSŤOVÁNÍ NAVIGAČNÍCH PRVKŮ NA STRÁNCE
SKIN	MOTIVY VZHLEDU STRÁNEK

Tabulka 6 Souborové přípony související s ASP.NET

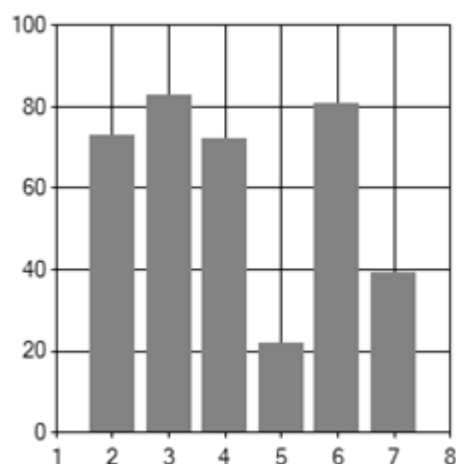
Adresářová struktura ASP.NET webové aplikace může být ovlivněna požadavky vývojáře. Kromě pár zarezervovaných adresářových názvů. Stránky se mohou roztřídit do množství adresářů. Taková struktura je typicky reflektována přímo v URL. Následující Tabulka 7 vypisuje zarezervované názvy adresářů se stručným popisem.

NÁZEV ADRESÁŘE	POPIS
APP_BROWSERS	OBSAHUJE SOUBORY SPECIFICKÉ WEBOVÝM PROHLÍŽEČŮM.
APP_CODE	OBSAHUJE ZDROJOVÉ KÓDY. SOUBORY V TOMTO ADRESÁŘI I JEHO PODADRESÁŘÍCH JSOU AUTOMATICKY ZKOMPILOVÁNY.
APP_DATA	ZÁKLADNÍ ADRESÁŘ PRO DATABÁZOVÉ SYSTÉMY.
APP_LOCALRESOURCES	OBSAHUJE LOKALIZOVANÉ STRÁNKY.
APP_GLOBALRESOURCES	SLOUŽÍ K UKLÁDÁNÍ PŘELOŽENÝCH ZPRÁV,

	KTERÉ SE VYSKYTUJÍ NA VÍCE NEŽ JEDNÉ STRÁNE.
APP	SESKUPUJE SOUBORY S ALTERNATIVNÍMI VZHLEDY STRÁNEK.
APP_WEBREFERENCE	SOUČÁSTÍ JSOU ODKAZY NA WEBOVÉ SLUŽBY
BIN	OBSAHUJE ZKOMPILOVANÝ KÓD (.DLL KNIHOVNY) PRO OVLÁDACÍ PRVKY NEBO KÓD NA KTERÝ SE ODKAZUJE Z WEBOVÉ APLIKACE.

Tabulka 7 Rezervovaná adresářová jména pro ASP.NET aplikaci

Programátoři ASP.NET využívají „User controls”, které na základě mechanismu událostí umožňují generování dynamických stránek. Nespokojí-li se programátor s „User controls” může si vytvořit vlastní „Custom controls”, které mají svůj kód kompilovány v knihovně „.dll” souboru. Protože ASP.NET v „User controls” s .NET framewroem 3.5 neobsahuje žádný nástroj na generování grafů, které jsou vhodné při zobrazení analýzy z log záznamů, tak bylo využito „Custom controls” z balíčku Microsoft Chart Controls. Po nainstalování programátor může využít grafů obsažených v souboru „System.Web.UI.DataVisualization.dll”. Grafy z Chart Controls fungují na principu generování „png” obrázku (obrázek s bezeztrátovou kompresí) tak, že každý vytvoření graf na ASP.NET stránce je v podstatě obrázkem s odkazem na soubor, který je uložen na web serveru.



Obrázek 7 Ukázka vygenerovaného grafu pomocí Microsoft Chart Controls

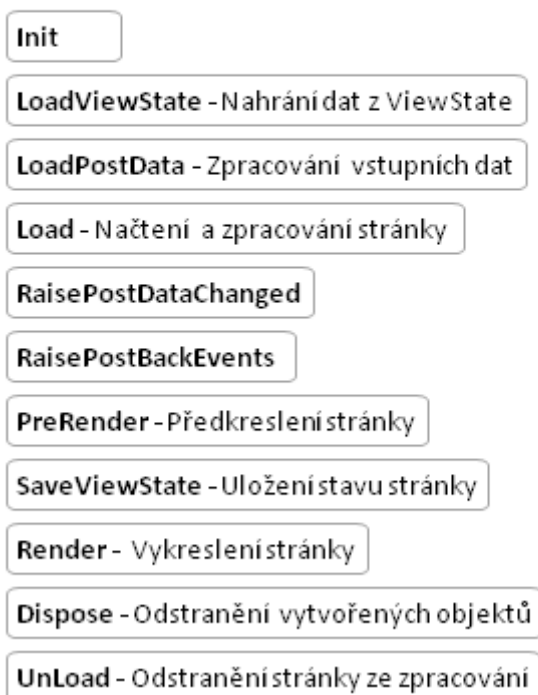
Výše uvedené událostmi řízené User Controls (ovládací prvky) vyžadují stavové prostředí, které webový protokol HTTP sám o sobě neposkytuje.[ASP.NET] ASP.NET proto tento problém řeší kombinací HTML a JavaScriptu pomocí dvou technik:

- ViewState - Uchovává informace mezi postbacky (opakované odesílání formuláře na server) v zakódovaném tvaru ve skrytém formulářovém poli. Výhoda tohoto řešení spočívá ve využití pouze HTML a nevyžaduje žádnou podporu na straně klienta ani serveru. Nevýhodou může být větší množství přenesených dat.
- Session State - Na rozdíl od ViewState uchovává veškeré stavové informace na straně serveru a to ve formě cookie (označuje data poslaná web serverem uložená na počítači klienta) nebo součástí URL (řetězec znaků identifikující zdroj informací). ASP.NET také umožňuje ukládání session state do databázového systému, které zachová stav i po restartu serveru.

Stránky prezentační technologie Log Pages využívají technologii ViewState, kvůli její jednoduchosti a možnosti připojení klienta, který má cookies ve webovém prohlížeči zakázané nebo nepodporované.

Znalost ASP.NET Page Life Cycle (životní cyklus stránky) je nezbytná pro programování stránek ASP.NET. Odesláním požadavku na server začíná životní cyklus stránky. Web server požadovanou stránku načte, provede zpracování a na závěr pošle klientovi zpět ve formě HTML stránky.

ASP.NET Life-Cycle



Obrázek 8 Schéma životní cyklu ASP.NET stránky

Životní cyklus začíná inicializací, tedy voláním události Init při současném zpracování funkce OnInit(). Dochází tím k inicializaci objektů na stránce. Stavové informace uložené ve ViewState jsou načteny v metodě LoadViewState. Metoda LoadPostData() obsluhuje aktualizaci odeslaného formuláře. Po zavolání události Load jsou všechny objekty na stránce inicializovány a prvky formuláře odpovídají prvkům zaslaným klientem. Právě v tomto okamžiku lze upravovat vlastnosti prvků na stránce, měnit hodnoty polí či si připravit data pro vykreslení grafů. Nastane-li se na stránce nějaká změna (např. vyplnění textového pole) dojde k oznámení o změně RaisePostDataChanged. Programátor je tímto informován, že původní stav prvku nesouhlasí se stavem současným.. Událost RaisePostBackEvents vyvolává události zaregistrované v metodě OnInit(). Avšak tuto událost mohou vyvolat pouze prvky s rozhraním „IPostBackDataHandler”. Poslední možnost pro programátora změnit vlastnosti prvků se nachází v události PreRender. Poté dojde k uložení stavu stránky SaveViewState do skrytého formulářového pole. Samotné vygenerování stránky a všech

obsažených prvků, které se mají zobrazit klientovi probíhá v metodě Render(). Metoda Dispose() odstraní objekty použité na stránce a na závěr událost UnLoad, která odstraní stránku ze zpracování a zašle klientovi.

Další technologií která je použita na Log Pages jsou tzv. Master Pages (hlavní stránky), které elegantním způsobem řeší opakující se části stránek (hlavičky nebo různé navigační prvky). Není tedy nutné pro každou stránku kopírovat stejnou hlavičku. Místo kopírování hlavičky na dané stránce odkážeme, kterou Master Page má stránka použít. Master Page potom obsahuje kód, který má být na všech stránkách stejný a dále „asp:ContentPlaceHolder” značí místo, kde bude umístěn obsah ostatních stránek. „asp:ContentPlaceHolder” se může na Master Page vyskytnout i vícekrát. Stránky s odkazem na Master Page poté musí obsahovat „asp:Content” s atributem „ContentPlaceHolderID” shodným s atributem „id” v tagu (značce) „asp:ContentPlaceHolder” na Master Page. Ilustrace této technologie je zachycena na Obrázku 9.

Log Pages						Master Page
Settings		Connection Statistics		Recent Logs		
type	event	ip	port	time	endtime	Content Page
N/A	N/A			22.6.2010 0:00:00		
N/A	N/A			22.6.2010 0:00:00		
N/A	N/A			22.6.2010 0:00:00		
N/A	N/A			22.6.2010 0:00:00		
N/A	N/A			22.6.2010 0:00:00		
N/A	N/A			22.6.2010 0:00:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:14:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:14:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:14:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:14:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:14:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:14:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:14:00		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:13:59		
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:08:05	11.5.2010 18:41:08	
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:08:05	11.5.2010 18:41:08	
Dundis Server	Client [dis]connect	162.168.1.20	800	11.5.2010 18:08:05	11.5.2010 18:41:08	

Obrázek 9 Využití Master Page v Log Pages

Poslední popsanou technologií bude komunikace mezi ASP.NET webovou aplikací a SQL databází. ASP.NET k tomuto účelu využívá hned několik přístupů.

Jedním z možných způsobů jak komunikovat s SQL serverem je využití jmenných prostorů „System.Data” a „System.Data.SqlClient”, obsažených v ADO.NET (Active Data Objects for .NET; umožňuje objektovou práci s relačními daty). Základní prvky ADO.NET:

- SqlDataReader - Dokáže stáhnout data z SQL databáze. Výhoda při použití jsou nízké paměťové nároky
- DataSet a DataTable - Dynamické struktury, které slouží k uchování dat.
- SqlDataAdapter - Zajišťuje přenos dat mezi DataSety a SQL serverem a jejich aktualizaci.

```
private static void ExecuteQuery(String query)
{
    SqlCommand Command = new SqlCommand(query);
    SqlConnection SQLconn = new SqlConnection(ConnString);
    Command.Connection = SQLconn;
    Command.Connection.Open();
    Command.ExecuteNonQuery();
    Command.Connection.Close();
}
```

Obrázek 10 Ukázka funkce pro vykonání SQL příkazu

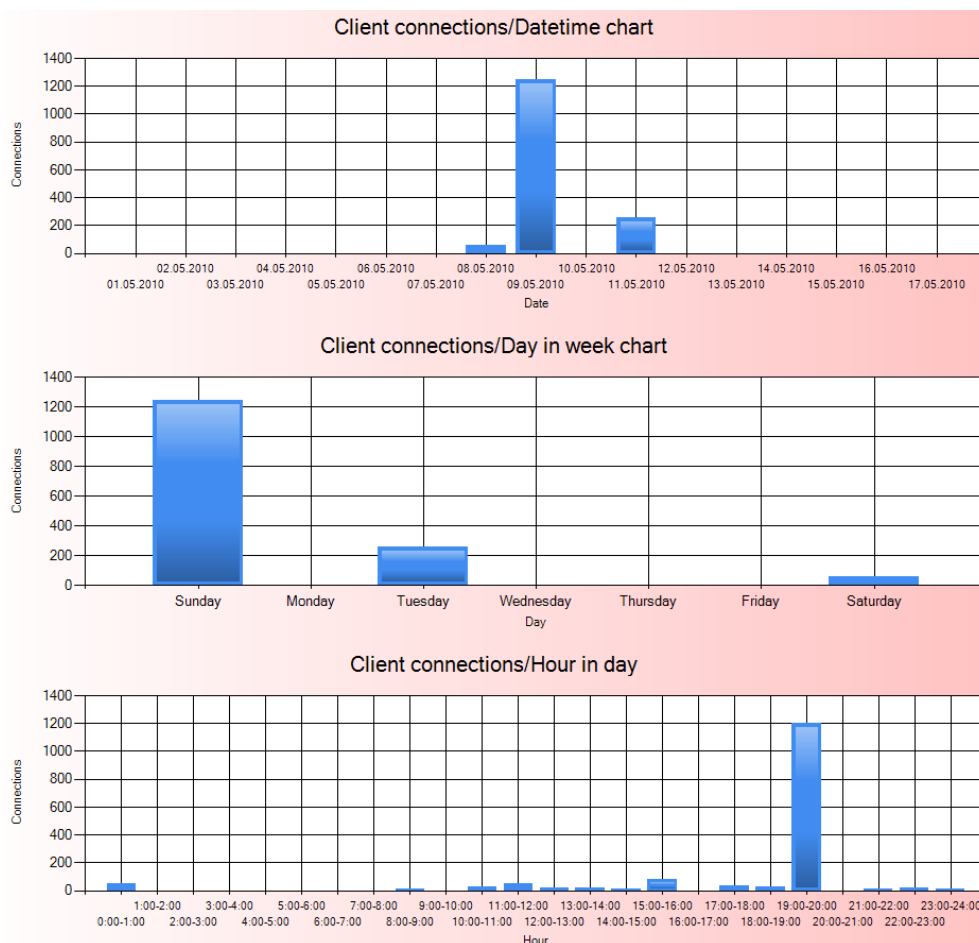
Metoda ExecuteQuery() k úspěšné komunikaci s SQL serverem vyžaduje parameter query (SQL příkaz) a tzv. „ConnectionString”. ConnectionString je řetězec znaků obsahující adresu SQL serveru, uživatelské jméno, heslo a název databáze, ke které se chceme připojit. ConnectionString podporuje celou řadu parametrů, jejich význam a využití lze nalézt v online MSDN knihovně.

Pomocí výše uvedených technologií lze sestavit interaktivní webovou aplikaci ASP.NET pro prezentaci analýzy log záznamů. Aplikaci jsem pojmenoval Log Pages a zpřístupnil všem uživatelům internetu. Log pages obsahují 3 stránky:

- LogViews.aspx - Hlavní stránka aplikace. Obsahuje podrobné statistiky log záznamů.
- Settings.aspx - Zde probíhá administrace log záznamů.
- RecentLogs.aspx - Slouží k rychlému zjištění nedávné aktivity.

Všechny uvedené statistiky jsou vztaženy ke zvolenému časovému horizontu a zvolené aplikaci (Log type). Statistiky přístupné na hlavní stránce:

- Overall - Zobrazí celkový počet připojených klientů a počet různých klientů (odlišení klientů probíhá na základě rozdílné IP adresy).
- IP statistics - Vypíše tabulku s IP adresami klientů a celkový počet připojení.
- Connections/Date - Graf závislosti celkového počtu připojení v jednotlivých dnech.
- Connections/Day in week - Graf zachycující připojení klientů v průběhu dní v týdnu.
- Connections/Hour in day - Graf připojení klientů v průběhu dne.
- Connections/Connection time - Informuje o počtu klientů vztaženého k délce připojení.



Obrázek 11 Ukázka vybraných grafů na stránce LogViews.aspx

4.6 DATABÁZOVÝ SYSTÉM

Databáze [16] (neboli datová základna) je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu (nejčastěji na pevném disku). V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými informacemi. Tento software se v české odborné literatuře nazývá systém řízení báze dat. Běžně se označením databáze myslí jak uložená data, tak i software.

Současný trh nabízí hned několik softwarů databázových systémů:

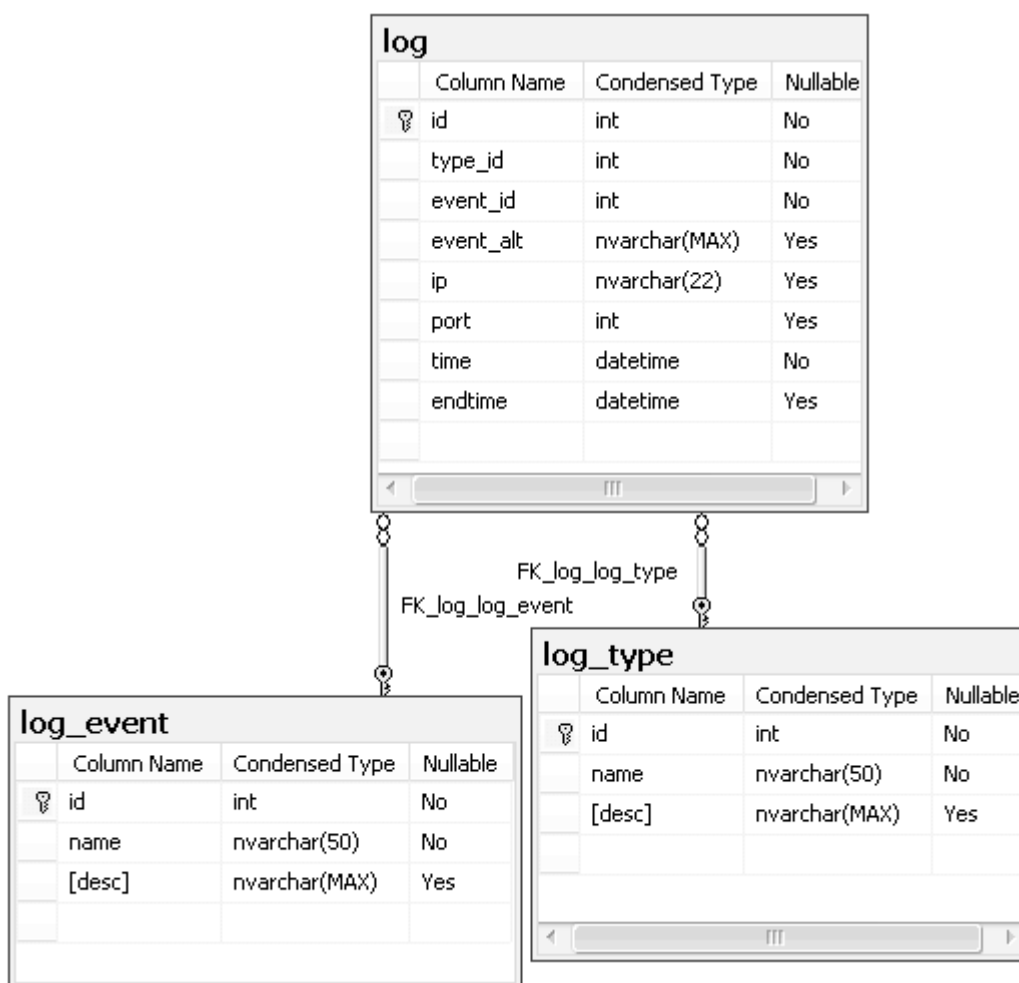
- MySQL - Nejrozšířenější Databázový systém díky otevřenosti jeho softwaru a licencování GPL i komerčnímu licencování.
- MSSQL - Alternativní řešení od Společnosti Microsoft.
- PostgreSQL - Další otevřený systém vyvíjený primárně pro Linux.
- Oracle - Moderní multiplatformní databázový systém od společnosti Oracle Corporation.

Protože celý systém pro analýzu a prezentaci dat je postaven na technologiích společnosti Microsoft, tak ani databázový systém netvoří výjimku. Zvolen byl Microsoft SQL Server 2005 (zkráceně MSSQL 2005) ve verzi Express, která je jedinou bezplatnou verzí, což s sebou přináší jistá omezení:

- Omezení velikosti jedné databáze na 4 GB
- Využití pouze jednoho procesoru
- Využití maximálně 1 GB operační paměti RAM
- Bez pokročilých možností a nástrojů vyšších verzí

Pro účely ukládání log záznamů do databáze bylo zapotřebí založit novou databázi pojmenovanou „LogDatabase” a vytvořit vhodnou strukturu propojených tabulek. Tvorba tabulek a jejich propojení probíhala v nástroji Microsoft SQL Server Management Studio Express, který je součástí instalace MSSQL 2005. Základní tabulka s názvem „log” v SQL databázi není nepodobná Tabulce 2 v kapitole 3.1 Analýza generovaných log souborů. Rozdíl spočívá v přidání sloupců: „id” (jedinečný identifikátor záznamu logu), „event_alt”(alternativní popis události) a „type_id”(slouží

k odlišení log záznamu různých aplikací). Sloupec „UDÁLOST” v Tabulce 2 je nahrazen sloupcem „event_id”, který spojuje tabulku „log” s tabulkou „log_event” v relaci 1:M (jeden záznam z tabulky „log_event” může být spojen s vícero záznamy v tabulce „log”). Stejně jako „event_id” tak i „type_id” spojuje tabulku „log” v relaci 1:M s další tabulkou, tentokrát ale s tabulkou „log_type”. Rozdělení tabulek odpovídá principům normalizace databáze. Celé schéma s názvy tabulek, sloupců včetně datového typu a možností prázdné hodnoty je zachyceno na následujícím Obrázku 12.



Obrázek 12 Diagram databáze LogDatabase

Jak může vypadat jeden záznam v tabulce „log” zachycuje Tabulka 7. Níže uvedený záznam má identifikátor „id” roven 1588. Pochází z aplikace pod označením 2 (v tomto konkrétním případě se jedná o Dundis server). Číslo 2 ve sloupci „event_id” oznamuje událost připojení a odpojení klienta. Klienta identifikuje sloupec „ip” a

„port”. Čas události je zachycen sloupcem „time”. Hodnota „NULL” v posledním sloupci říká že klient ještě nebyl odpojen, nebo došlo k výpadku serveru a čas odpojení klienta není zaznamenán.

ID	TYPE_ID	EVENT_ID	EVENT_ALT	IP	PORT	TIME	ENDTIME
1588	2	2	NULL	192.168.1.1	800	2010-05-11 18:08:05	NULL

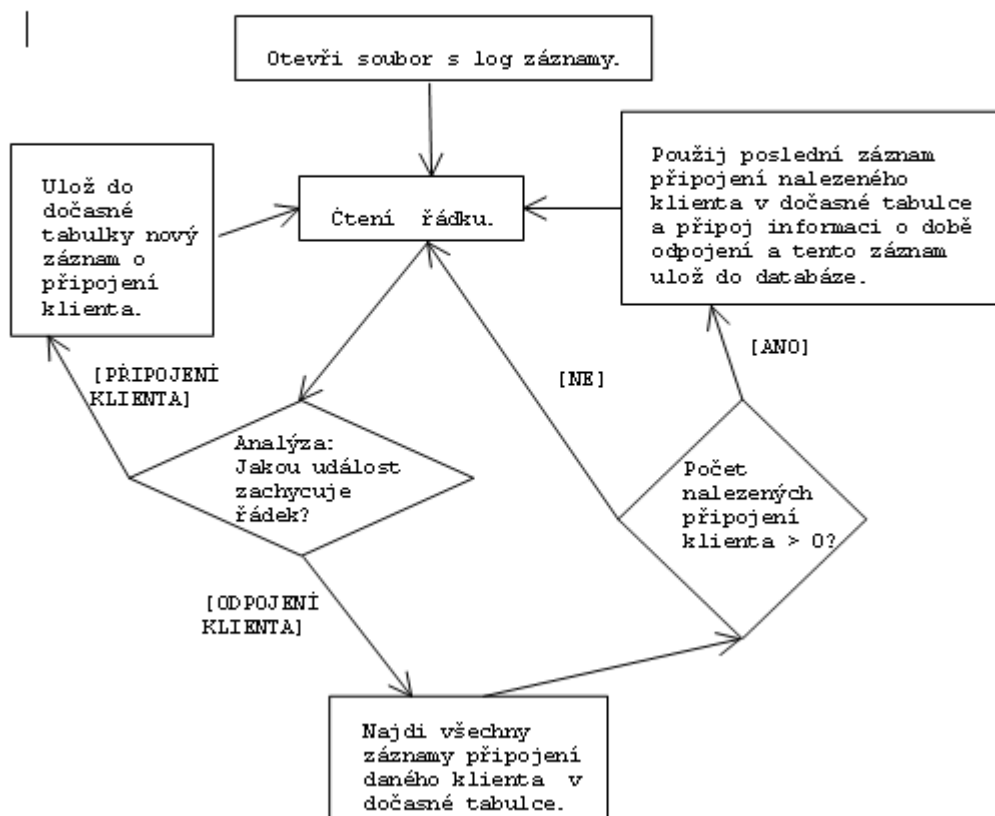
Tabulka 8 Jeden záznam logu v tabulce „log”

5 TECHNOLOGIE ZDROJE DAT

K funkčnosti celého systému chybí jen zápis log záznamů z aplikací do SQL databáze. Po analýze situace aplikace a jejich log záznamů se jeví jako řešení dvě možné varianty. První varianta spočívá ve vytvoření programu, který k log záznamu přistoupí přímo a řádek po řádku analyzuje data podle naprogramovaného algoritmu a výsledek zapíše do databáze. Výhoda spočívá v tom, že není zapotřebí žádného zásahu do aplikace generující log soubor, ale na druhou stranu je nutno počítat s monitorováním změn v souboru pro aktualizaci dat v databázi. Druhá varianta zahrnuje vytvoření hlavičkového souboru a zásah do zdrojového kódu aplikace. V hlavičkovém souboru potom budou nadefinovány funkce pro přímý zápis „log” záznamu do databáze. To znamená, že kromě možnosti zápisu vlastního log souboru bude aplikace mít funkci zápisu logu přímo do databáze. Výhodou oproti prvnímu přístupu je aktuálnost dat log záznamů přístupných online a menší výpočetní náročnost.

Pro vyzkoušení první metody byla naprogramována jednoduchá konzolová aplikace „LogWriter”, která po spuštění předem nadefinovaný log soubor otevře, zanalyzuje a výsledek zapíše do „LogDatabase”. Schéma algoritmu zachycuje Obrázek 13.

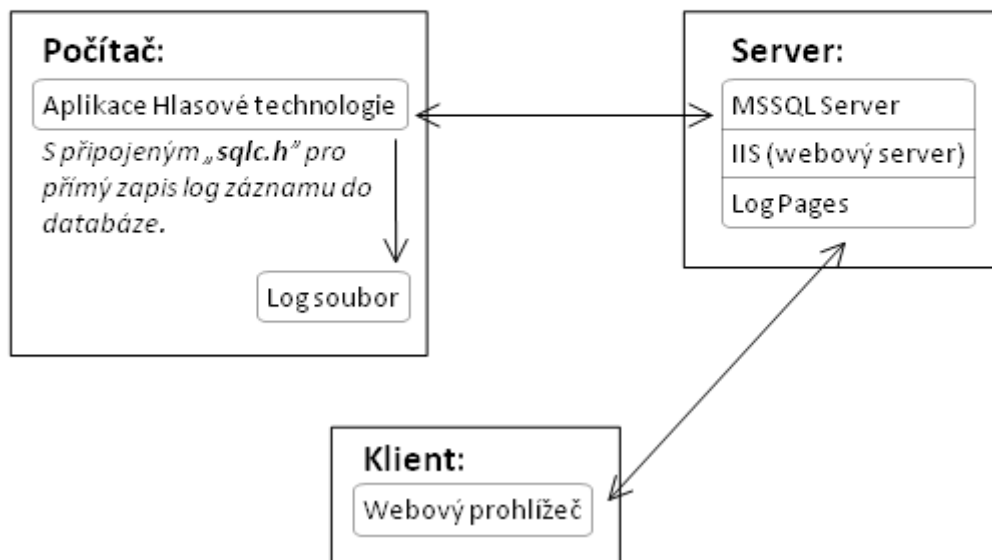
Schéma programu LogWriter



Obrázek 13 Schéma programu LogWriter

Druhý přístup se liší od prvního v tom, že nedochází k žádnému čtení řádku, které je nahrazeno metodou pro zápis do databáze. Metoda se potom volá se všemi důležitými parametry pro záznam logu. Druhý přístup je zpracován jako hlavičkový soubor s názvem „sqlc.h” napsaným v jazyce C++, který obsahuje kód pro komunikaci s MSSQL databází a který voláním výše uvedené metody provede zaznamenání log záznamu do databáze. V hlavičkovém souboru se potom nastaví parametry pro připojení („ConnectionString”) a typ aplikace, který musí být shodný se záznamem v tabulce „log_types”. Hlavičkový soubor „sqlc.h” se poté přilinkuje k aplikaci generující „log” soubory (viz. Obrázek 14) a v místě volání funkce pro zápis do „log” souboru se přidá funkce pro zápis „log” záznamu do databáze pojmenovanou „SendLogToDB”.

Schéma systému s využitím přímého zápisu



Obrázek 14 Schéma systému s využitím přímého zápisu do databáze

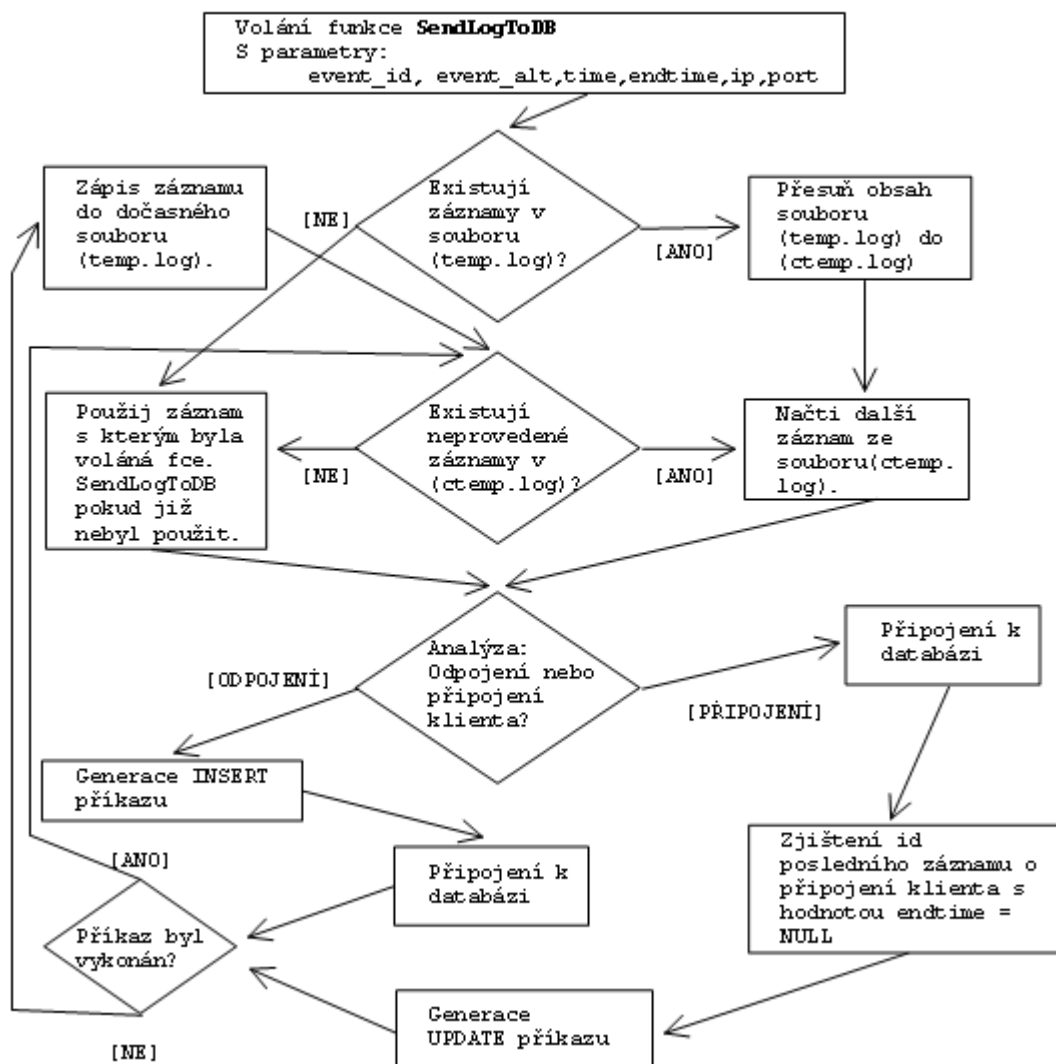
Funkce SendLogToDB pomocí parametrů s kterými byla volána sestaví SQL příkaz, kterým log záznam nahraje do databáze. Mohou nastat 2 situace, které se budou logovat::

- Připojení klienta - V případě připojení klienta se jedná o jednodušší variantu. Použije se „INSERT” SQL příkaz, který log záznam zapíše do databáze.
- Odpojení klienta - V této situaci je zapotřebí zjistit, kdy se daný uživatel připojil, aby mohl být doplněn záznam o odpojení v tabulce „log”. Algoritmus se zeptá databáze na poslední záznam dané kombinace IP adresy a portu a vrátí identifikátor „id” odpovídajícího záznamu, který je následně použit pro „UPDATE” SQL příkaz doplňující čas odpojení klienta v databázi.

Pokud se připojení k databázi nezdaří, potom je log záznam uložen na disk do dočasného souboru ve formátu CSV(hodnoty oddělené čárkou). Při dalším volání funkce SendLogToDB se nejprve zjistí jestli existují nějaké neodeslané záznamy a pokud ano tak se je pokusí zapsat do databáze. V případě úspěchu se obsah dočasného

souboru promaže avšak v opačném případě se nový log záznam připsá na konec dočasného souboru („temp.log“). Podrobné schéma funkce „SendLogToDB“ zachycuje Obrázek 15. Podrobnější informace k použití této funkce lze nalézt v komentářích v souboru „sqlc.h“ na přiloženém disku.

Schéma funkce SendLogToDB



Obrázek 15 Schéma funkce SendLogToDB

6. ZÁVĚR

Cílem bakalářské práce bylo navrhnout systém pro statistickou analýzu a prezentaci „log” souborů z aplikací hlasových technologií na pracovišti školitele. Statistická analýza probíhá na základě příslušných SQL dotazů do databáze, kde jsou uloženy „log” záznamy. SQL dotaz potom vrátí hodnoty, které jsou prezentovány na stránkách Log Pages. O zapisování dat do databáze se stará funkce v hlavičkovém souboru „sqlc.h”, který je nalinkován v aplikacích generujících „log” soubory.

Při zpracování této bakalářské práce bylo nezbytné seznámit se s dostupnými technologiemi pro zpracování a prezentaci textových souborů. K realizovaní bylo použito platformy .NET od společnosti Microsoft, především kvůli její robustnosti, nepřehlednému množství odborných textů a přístupnosti. Práce zasahovala hned do několika vývojových prostředí. Jedná se o prostředí Visual Studio, kde pomocí ASP.NET a C# je vytvořená samotná webová aplikace Log Pages. Dále prostředí MSSQL serveru, kde se nelze obejít bez základní znalosti syntaxe SQL dotazů a nakonec samotné prostředí Windows. V neposlední řadě byl použit jazyk C++ při tvorbě hlavičkového souboru pro přímý zápis log záznamu do databáze.

Celý systém prezentuje statistiky online avšak s jeho výkonem nejsem zcela spokojen. Proto zde existuje další prostor pro vylepšení, optimalizaci a rozšíření funkčnosti, které může být námětem pro další ročníkové projekty nebo závěrečné práce.

Věřím, že prezentovaná data poskytnutá stránkami Log Pages přinesou cenné informace pro další vývoj nejenom hlasových technologií.

SEZNAM POUŽITÉ LITERATURY

- [1] HORVÁTH, Tomáš. *Teoretický úvod do relačních databází* [online]. 8.11.2007. URL:<<http://programujte.com/index.php?akce=clanek&cl=2007110801-teoreticky-uvod-do-relacnich-databazi>>
- [2] PÍSEK, Slavoj. *ASP.NET začínáme programovat*. Grada, 2003. 228 s.ISBN 80-247-0526-5.
- [3] Wikipedia contributors. *Tabulkový procesor* [online]. 31.1.2010. URL:<http://cs.wikipedia.org/wiki/Tabulkov%C3%BD_procesor>
- [4] MSDN Library. *SqlConnection Class* [online]. 2010. URL:<<http://msdn.microsoft.com/en-us/library/sd2728ad%28v=VS.100%29.aspx>>
- [5] Wikipedia contributors. *NET Framework* [online]. 25.4.2005. URL:<http://en.wikipedia.org/wiki/.NET_Framework>
- [6] MOZDŘEŇ, Karel. *C# a ASP.NET* [online]. 17.2.2009. URL:<http://homel.vsb.cz/~moz017/c-sharp-asp-net/UcebniceASP_NET.pdf>
- [7] Neznámý autor. *Laboratoř počítačového zpracování řeči* [online]. 1.5.2010. URL:<<https://www.ite.tul.cz/speechlab/index.php>>
- [8] Neznámý autor. *Internetový řečový rozpoznávač* [online]. 1.5.2010. URL:<<https://www.ite.tul.cz/speechlab/index.php/stari-projekty/dundis.html>>
- [9] Wikipedia contributors. *Internet Protocol* [online]. 26.4.2010. URL:<http://cs.wikipedia.org/wiki/Internet_Protocol>
- [10] Wikipedia contributors. *IP adresa* [online]. 21.3.2010. URL:<http://cs.wikipedia.org/wiki/IP_adresa>
- [11] Wikipedia contributors. *Port* [online]. 6.3.2009. URL:<http://cs.wikipedia.org/wiki/S%C3%AD%C5%A5ov%C3%BD_port>
- [12] Wikipedia contributors. *TCP* [online]. 17.5.2010. URL:<<http://cs.wikipedia.org/wiki/TCP>>
- [13] Wikipedia contributors. *UDP* [online]. 17.5.2010. URL:<<http://cs.wikipedia.org/wiki/UDP>>
- [14] GROTE, Marc. *Using the Logparser Utility to Analyze Exchange MS Logs* [online]. 26.6.2006. URL:<<http://www.msexchange.org/tutorials/Using-Logparser-Utility-Analyze-ExchangeIIS-Logs.html>>
- [15] Wikipedia contributors. *ASP.NET*. 23.5.2010. URL:<<http://cs.wikipedia.org/wiki/ASP.NET>>
- [16] Wikipedia contributors. *Databáze*. 23.5.2010. URL:<<http://cs.wikipedia.org/wiki/Datab%C3%A1ze>>

SEZNAM OBRÁZKŮ

Obrázek 1 Ukázka log souboru ze serveru Dundis	15
Obrázek 2 Komunikace mezi PC a Serverem	17
Obrázek 3 Schéma prezentační technologie.....	25
Obrázek 4 Správa IIS 6.0.....	30
Obrázek 5 Přilinkování modelu Code-Behind v souboru Settings.aspx	31
Obrázek 6 Ukázka souboru Settings.aspx.cs s dynamickým kódem.....	31
Obrázek 7 Ukázka vygenerovaného grafu pomocí Microsoft Chart Controls	33
Obrázek 8 Schéma životní cyklu ASP.NET stránky	35
Obrázek 9 Využití Master Page v Log Pages	36
Obrázek 10 Ukázka funkce pro vykonání SQL příkazu.....	37
Obrázek 11 Ukázka vybraných grafů na stránce LogViews.aspx	38
Obrázek 12 Diagram databáze LogDatabase	40
Obrázek 13 Schéma programu LogWriter	43
Obrázek 14 Schéma systému s využitím přímého zápisu do databáze	44
Obrázek 15 Schéma funkce SendLogToDB.....	45

PŘÍLOHY

OBSAH DVD

Zdrojové kódy:

- LogWriter (C#)
- Hlavičkový soubor „sqlc.h” (C++)
- LogPages webová aplikace (ASP.NET, C#)
- Prázdná databáze s vytvořenými tabulkami (MSSQL 2005)

SOFTWARE POTŘEBNÝ PRO SPUŠTENÍ APLIKACE

- Operační systém Windows XP nebo vyšší
- Microsoft .NET 2.0 SP2 a .NET 3.5
- MSSQL server 2005
- MS Chart Controls

PODPOROVANÉ VERZE PROHLÍŽEČŮ

- Internet Explorer 6
- Mozilla Firefox verze 3.0 a vyšší